Decentralized Algorithms for Operating Coded Wireless Networks

Fang Zhao, Desmond S. Lun, Muriel Médard, and Ebad Ahmed
Laboratory for Information and Decision Systems
Massachusetts Institute of Technology, Cambridge, MA 02139, USA
E-mail: {zhaof, dslun, medard, ebad}@mit.edu

***Abstract*— The problem of subgraph optimization for multicast connections with network coding can be solved using the subgradient method. In this paper, we focus on the problem of min-energy multicast in both static and dynamic multi-hop wireless networks. We take this optimization problem and construct different candidate algorithms for solving it in a distributed manner. Our simulations show that the subgradient method is robust to network changes, and yields significant energy savings in multicasts as compared to cases where only routing is allowed. Moreover, the distributed method can be easily extended to lossy networks.**

I. Introduction

When network coding is used to perform multicast, the problem of establishing minimum-cost multicast connection is equivalent to two effectively decoupled problems [1]: one of determining the subgraph to code over, and the other of determining the code to use over that subgraph. The latter problem has been studied in [2], [3], [4], [5], and a variety of methods have been proposed, which include employing simple random linear coding at every node. Such random linear coding schemes are completely decentralized, requiring no coordination between nodes, and can operate under dynamic conditions [6].

The former problem has been studied in [7], [8]. While [7] looked at centralized solutions, [8] proposed approaches to find minimum-cost multicast subgraphs for both linear and strictly convex cost functions in a decentralized manner.

The presentation of the subgradient method in [8], however, only states the algorithm and proves its convergence. In this paper, we study the implementation issues of this algorithm. We focus on multi-hop wireless networks because, aside from wireless networks being a prime application for network coding, our model for wireless networks subsumes wireline networks.

Wireless networks are also generally more challenging than wireline ones. One of the challenges of wireless networks, such as ad hoc networks and sensor networks, is variability of the network topology. Topology changes can be caused by mobility of users, sleeping or waking up of nodes, or the shadowing effect due to moving obstacles. We show that the subgradient method is robust to topology changes, and nodes are able to adjust their transmission power levels to move smoothly and quickly to a new optimal subgraph in a distributed manner.

Simulation results show that the subgradient method converges to optimal solutions quickly and that it is robust to network changes. In the past, we have seen successful translation of robust distributed optimization algorithms, such as distributed Bellman-Ford (DBF), into popular protocols, such as the Open Shortest Path First (OSPF) protocol. We hope that the subgradient method will lead to protocol design for subgraph optimization in coded networks in the future.

The remaining part of this paper is organized as follows. Section II describes the network model used and reviews the formulation of the subgraph optimization problem. In Section III, we first present the canonical subgradient method for decentralized subgraph optimization and then analyze its complexity. We propose various modifications to improve the convergence rate of the canonical algorithm under static and dynamic networks. Simulation results are presented in Section IV to illustrate the effectiveness of these methods, and we conclude in Section V.

II. Network model and problem setup

We model a wireless networks with a directed hypergraph $\mathcal{H} = (\mathcal{N}, \mathcal{A})$, where $\mathcal{N}$ is the set of nodes and $\mathcal{A}$ is the set of hyperarcs. A hypergraph is a generalization of a graph, where, rather than arcs, we have hyperarcs. A hyperarc is a pair (i, J), where i, the start node, is an element of $\mathcal{N}$ and J, the set of end nodes, is a non-empty subset of $\mathcal{N}$. Each hyperarc (i, J) represents a broadcast link from node i to nodes in the non-empty set J (e.g. Fig. 1). This link may be lossless or lossy, i.e. it may or may not be subject to packet erasures. Let A_{iJK} be the counting process describing the arrival of packets that are injected on hyperarc (i, J) and received by exactly the set of nodes $K \subset J$, i.e. for $\tau \geq 0$, $A_{iJK}(\tau)$ is the total number of packets that are injected on hyperarc (i, J) and received by all nodes in K (and no nodes in $\mathcal{N} \setminus K$) between time 0 and time τ. We assume that A_{iJK} has an average rate z_{iJK}; more precisely, we assume that $\lim_{\tau\to\infty} A_{iJK}(\tau)/\tau = z_{iJK}$ almost surely. When links are lossless, we have $z_{iJK} = 0$ for all non-empty $K \subsetneq J$. Let $z_{iJ} := \sum_{K\subset J} z_{iJK}$ be the average rate at which packets are injected into hyperarc (i, J). The rate vector z, consisting of z_{iJ} for $(i, J) \in \mathcal{A}$, is the coding subgraph.

We associate with hyperarc (i, J) the non-negative number a_{iJ}, which represents the cost per unit rate of traffic injected into the hyperarc. Hence the cost of hyperarc (i, J) is $a_{iJ} z_{iJ}$.

1-4244-1564-0/07/$25.00 ©2007 IEEE

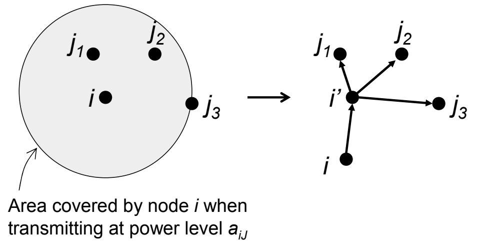

Fig. 1. Hyperarc (i, J), where $J = \{j_1, j_2, j_3\}$. Nodes j_1, j_2, and j_3 can all be reached at the same time with cost a_{iJ}, and this is equivalent to having three unit capacity links from i to j_1, j_2, and j_3. Here, we also include a virtual unit-capacity link (i, i') to impose the constraint that information transmitted on the three links must be the same.

We assume in this paper that the cost reflects energy consumption. We justify this assumption on the basis that we take energy as the most significant constraint, so there are, for example, sufficient time or frequency slots to guarantee that no two transmissions ever interfere.

It is shown in [9], [10], [11] that a subgraph z is capable of supporting a multicast connection of rate R from source node s to a set of terminal nodes T if and only if the max-flow from s to any $t \in T$ is greater than or equal to R. Hence, the problem of min-energy multicast can be posed as the following LP problem [12]:

$$\begin{aligned}
\min \quad & \sum_{(i,J)\in\mathcal{A}} a_{iJ} z_{iJ} \\
\text{s.t.} \quad & \sum_{j\in K} x_{iJj}^{(t)} \le \sum_{\{L\subset J | L\cap K \neq \emptyset\}} z_{iJL}, \\
& \qquad \forall\ (i,J)\in\mathcal{A},\ K\subset J,\ t\in T, \\
& \sum_{\{J|(i,J)\in\mathcal{A}\}} \sum_{j\in J} x_{iJj}^{(t)} - \sum_{\{(j,I)|(j,I)\in\mathcal{A}, i\in I\}} x_{jIi}^{(t)} = \sigma_i^{(t)}, \\
& \qquad \forall i\in\mathcal{N}, t\in T, \\
& x_{iJj}^{(t)} \ge 0, \qquad \forall\ (i,J)\in\mathcal{A},\ j\in J,\ t\in T,
\end{aligned} \tag{1}$$

where

$$\sigma_i^{(t)} = \begin{cases} R & \text{if} \quad i = s, \\ -R & \text{if} \quad i = t, \\ 0 & \text{otherwise.} \end{cases}$$

We consider multicast connections because they are the most general type of connection—subsuming unicast and broadcast as special cases.

A simplification can be made if we assume that the network is lossless, because then we have $z_{iJK} = 0$ for all non-empty $K \subsetneq J$. A further simplification arises if we assume that, when nodes transmit in a lossless network, they reach all nodes in a certain area that is an increasing function of their transmission power. So, if node i has M_i power levels that induce broadcast links represented by the hyperarcs $(i, J_1), (i, J_2), \dots, (i, J_{M_i})$ with increasing power, then $J_1 \subsetneq J_2 \subsetneq \cdots \subsetneq J_{M_i}$. (We assume that there are no identical links, as duplicate links can effectively be treated as a single link.) Let $J_1^{(i)}, J_2^{(i)}, \dots, J_{M_i}^{(i)}$ be the M_i sets of nodes reached by node i with its M_i power levels. Then, for $(i,j) \in \mathcal{A}' := \{(i,j)|(i,J) \in \mathcal{A}, J \ni j\}$, we introduce the variables $\hat{x}_{ij}^{(t)} := \sum_{m=m(i,j)}^{M_i} x_{iJ_m^{(i)}j}^{(t)}$, where $m(i,j)$ is the unique m such that $j \in J_m^{(i)} \setminus J_{m-1}^{(i)}$. Now problem (1) can be reformulated as the following optimization problem, which has substantially fewer variables.

$$\begin{aligned}
\text{minimize} \quad & \sum_{(i,J)\in\mathcal{A}} a_{iJ} z_{iJ} \\
\text{subject to} \quad & \sum_{k\in J_{M_i}^{(i)} \setminus J_{m(i,j)-1}^{(i)}} \hat{x}_{ik}^{(t)} \le \sum_{n=m(i,j)}^{M_i} z_{iJ_n^{(i)}}, \\
& \qquad \forall\ (i,j)\in\mathcal{A}', t\in T, \\
& \hat{x}^{(t)} \in F^{(t)}, \\
& \qquad \forall\ t\in T,
\end{aligned} \tag{2}$$

where $\hat{F}^{(t)}$ is the bounded polyhedron of points $\hat{x}^{(t)}$ satisfying the conservation of flow constraints

$$\sum_{\{j|(i,j)\in\mathcal{A}'\}} \hat{x}_{ij}^{(t)} - \sum_{\{j|(j,i)\in\mathcal{A}'\}} \hat{x}_{ji}^{(t)} = \sigma_i^{(t)},$$
$$\forall i\in\mathcal{N}, \text{ and } \hat{x}_{ij}^{(t)} \ge 0, \forall\ (i,j)\in\mathcal{A}'.$$

We focus on this particular formulation in the rest of this paper. Our considerations can, however, be modified in a straightforward manner to apply to the general formulation in (1). To perform the optimization in a distributed manner, the subgraph optimization scheme uses the Lagrangian dual of (2) given below.

$$\begin{aligned}
\text{maximize} \quad & \sum_{t\in T} q^{(t)}(p^{(t)}) \\
\text{subject to} \quad & \sum_{t\in T} p_{iJ_m^{(i)}}^{(t)} = s_{iJ_m^{(i)}}, \quad \forall\ i\in\mathcal{N}, m = 1,\dots,M_i, \\
& p_{iJ}^{(t)} \ge 0, \qquad \forall\ (i,J)\in\mathcal{A}, t\in T,
\end{aligned} \tag{3}$$

where

$$s_{iJ_m^{(i)}} := a_{iJ_m^{(i)}} - a_{iJ_{m-1}^{(i)}},$$

$$q^{(t)}(p^{(t)}) := \min_{\hat{x}^{(t)}\in\hat{F}^{(t)}} \sum_{(i,j)\in\mathcal{A}'} \left(\sum_{m=1}^{m(i,j)} p_{iJ_m^{(i)}}^{(t)} \right) \hat{x}_{ij}^{(t)}.$$

III. Subgradient method for decentralized subgraph optimization

The subgraph optimization scheme tries to converge to the optimal solution of problem (2) by using subgradient optimization methods on the dual problem (3). We first present the canonical subgradient methods for distributed subgraph optimization here, and in the subsequent subsections, we present complexity analysis, and various heuristics to improve the convergence performance of the canonical algorithms in both static and dynamic networks. Here, computation takes place at each node, which needs only to be aware of the costs of its incoming and outgoing links.

In the standard subgradient method, we start with a feasible solution $p[0]$, and in the nth iteration, use $p^{(t)}[n]$ values as the hyperarc costs, and run a distributed shortest path algorithm (e.g., DBF) to determine $x^{(t)}[n]$ for all $t \in T$. We then calculate a subgradient of the dual function with respect to $p^{(t)}_{iJ^{(i)}_m}[n]$, namely $g^{(t)}_{iJ^{(i)}_m}[n] := \sum_{k \in J^{(i)}_{M_i} \setminus J^{(i)}_{m-1}} \hat{x}^{(t)}_{ik}[n]$. Then $p[n+1]$ is updated as

$$p[n+1] := \left[p[n] + \theta[n] \sum_{t \in T} g^{(t)}[n] \right]^+_P,$$

where $[\cdot]^+_P$ denotes the projection onto the constraint set P in (3), and $\theta[n]$ is a positive stepsize. The projection can be done in a distributed manner as given in [8].

At the end of each subgradient iteration, nodes recover a primal solution $\tilde{x}[n]$ from the previous results and compute $z[n]$ based on $\tilde{x}[n]$. More details on the recovery of primal solution will be presented in Section III-B. Since any set of z values obtained from an intermediate $\tilde{x}[n]$ gives a feasible subgraph on which networking coding can be used to perform the multicast, we do not have to wait till the algorithm has converged to start the multicast. Instead, it can be started right after the first iteration, and the subgraph will then be adjusted slowly to be more efficient in energy consumption.

A. Complexity analysis

Assume the network is synchronous, the time complexity for one iteration of the standard subgradient method is $O(n)$, which is the same as running a DBF, since the N_T DBF sessions can be run in parallel. As for message complexity, an iteration in the standard method takes $O(n|A|N_T)$ messages. As for the total number of iterations, it is shown in Section IV that after a small number of iterations, $z[n]$ gives a subgraph that is fairly close to the optimal solution in terms of total cost.

B. Initialization and primal solution recovery

Using the standard subgradient method framework, we propose several methods for initializing the dual vector $p[0]$, and for recovering primal solutions $\{\tilde{x}[n]\}$, for both static and dynamic networks.

1) Static networks: We start with static networks, where the topology of the network is fixed throughout the multicast. We first introduce a naive way of initializing the dual variables.

- **Averaging method** — The simplest way to generate feasible initial values for the dual variables is to assign $p^{(t)}_{iJ} = s_{iJ}/N_T$ for all $t \in T$ and all $(i, J) \in \mathcal{A}$. This method is useful in static networks since no prior information of the multicast problem is available at the nodes.

While the subgradient optimization method yields good approximations of the optimal value of the dual problem (3) after sufficient number of iterations, it does not necessarily yield a primal optimal solution. To recover primal solutions in the subgradient optimization, we propose the following two options.

- **Original primal recovery** — This method is from [8] and [13]. Define the sequence $\{\tilde{x}[n]\}$ as $\tilde{x}[n] = \frac{1}{n}\sum_{l=1}^{n} \hat{x}[l]$. It can be shown that an accumulation point of the sequence of primal iterates $\{\tilde{x}[n]\}$ is an optimal solution to the primal problem (2) if the step sizes $\{\theta[n]\}$ are chosen properly [13].

- **Modified primal recovery** — Using the original primal recovery method, we observed in simulations that the cost of the multicast starts at a high value, and then converges slowly to the optimal value through iterations. One reason for the slow convergence is that it is recovered by averaging $\hat{x}[n]$ values from all the iterations. The effect of the first few high cost iterations takes a large number of iterations later to dilute. A heuristic way to improve the convergence rate is to discard these "bad" primal solutions after some time, and just average over the most recent N_a number of iterations in primal solution recovery.

2) Dynamic networks: As opposed to the static assumption, many wireless networks have topologies that are dynamic. Whenever a topology change occurs, we need to restart the distributed algorithm, as the subgraph used for multicast before the topology change might have become infeasible. In such cases, all the methods discussed in the previous subsection for dual variable initialization and primal solution recovery are still applicable. However, since the difference between the optimal solutions to the multicast problem before and after the changes are usually small, we should make use of the solutions $\bar{x}$ and $\bar{p}$ before the topology changes in the new iterations to improve the convergence rate. We propose additional methods to initialize p and update $\tilde{x}$, that make use of this old information.

For dual variable initialization, we present two additional heuristics.

- **Scaling method** — In this method, each node i scans through its set of outgoing hyperarcs $(i, J_1), (i, J_2), \ldots, (i, J_{M_i})$. For hyperarc (i, J_m), find an existing hyperarc $(i, \bar{J}_{\bar{m}})$ before the topology change such that there exists j satisfying $m(i, j) = m$ and $\bar{m}(i, j) = \bar{m}$. We scale the $\{\bar{p}^{(t)}_{i\bar{J}_{\bar{m}}}\}$ values so that they satisfy the new dual constraints. Specifically, setting $\bar{s}_{i\bar{J}_{\bar{m}}} := \sum_{t \in T} \bar{p}^{(t)}_{i\bar{J}_{\bar{m}}}$, we assign $p^{(t)}_{iJ_m} := \bar{p}^{(t)}_{i\bar{J}_{\bar{m}}} \times s_{iJ_m}/\bar{s}_{i\bar{J}_{\bar{m}}}$. If no such hyperarc $(i, \bar{J}_{\bar{m}})$ exists, we simply use $p^{(t)}_{iJ_m} := s_{iJ_m}/N_T$.
- **Projection method** — In this method, we use an intermediate $\tilde{P}$. For hyperarc (i, J_m), we again find an existing hyperarc $(i, \bar{J}_{\bar{m}})$ before the topology change such that there exists j satisfying $m(i, j) = m$ and $\bar{m}(i, j) = \bar{m}$. We set $\tilde{p}^{(t)}_{iJ_m} := \bar{p}_{i\bar{J}_{\bar{m}}}$ if such a hyperarc $(i, \bar{J}_{\bar{m}})$ exists, and we set $\tilde{p}^{(t)}_{iJ_m} := 0$ otherwise. We can then project this $\tilde{p}$ onto the new feasible region of the dual problem using the method presented in [8] to obtain an initial point p for the decentralized algorithm.

On the primal side, we observe that as long as there is no

$(i,j) \in \bar{\mathcal{A}}'$ such that $\bar{x}_{ij}^{(t)} > 0$ and $(i,j) \notin \mathcal{A}'$, then the old $\{\hat{x}[n]\}$ values from the previous iterations are still valid under the new topology. Thus, they can be used in the recovery of the current primal optimal solution. Based on this observation, we propose the following heuristic for primal solution recovery.

- **Look-back primal recovery** — When a topology change occurs, each node checks if there exists $(i,j) \in \bar{\mathcal{A}}'$ such that $\bar{x}_{ij}^{(t)} > 0$ and $(i,j) \notin \mathcal{A}'$. If yes, it sends out a signal to all nodes, and $\tilde{x}[n]$ is computed based only on the new $x[n]$ as in the original or modified primal recovery method above. On the other hand, if no link is removed, the averaging is done over N_a iterations before and after the topology change. The assumption that nodes can be informed of the removal of an active link within one iteration is reasonable, since, in each iteration, DBF is used to compute $x[n]$ and sending such a signal to all nodes should take less time than running DBF.

IV. Simulation results

A. Static networks

Simulations of the subgraph optimization scheme in static networks showed that the standard subgradient method has a better convergence time as compared to the incremental subgradient method, thus, in this section, we only present results for the standard method. We set up random wireless networks in a 10×10 square with a radius of connectivity $r = 3$. We consider multicast connections of unit rate. Fig. 2 shows the average convergence performance for the proposed algorithms for networks with 30/50 nodes and 4/8 terminals in the multicast. The step sizes used in the subgradient method are $\theta[n] = n^{-\alpha}$ with $\alpha = 0.8$ for $n = 0, 1, \cdots$. For the modified primal recovery method, the parameter N_a is set to 30. As we can see, the two primal cost curves coincide for the first 30 iterations, and after that, the modified method converges to the optimal value faster than the original method.

To compare the performance of the proposed scheme to the cost of multicast when network coding is not used, we use the Multicast Incremental Power (MIP) algorithm described in [14], which is a centralized heuristic algorithm to perform minimum-energy multicast in wireless networks. For the same setting, the average cost values for the multicast given by MIP algorithm are also shown in Fig. 2. As can be seen, in both cases, even the initial high cost values from our distributed algorithms are lower than that from the centralized MIP algorithm. Moreover, in fewer than 50 iterations, the cost of the multicast using modified primal recovery is within 5% higher than the optimal value. Therefore, in a small number of iterations, the decentralized subgraph optimization algorithms yield solutions to the multicast problem with energy significantly lower than that for multicast without network coding even if a centralized scheme is used.

B. Mobile networks

To illustrate the performance of our algorithms in dynamic networks, we use random networks with mobile nodes. The mobility model used in our simulations is the Random

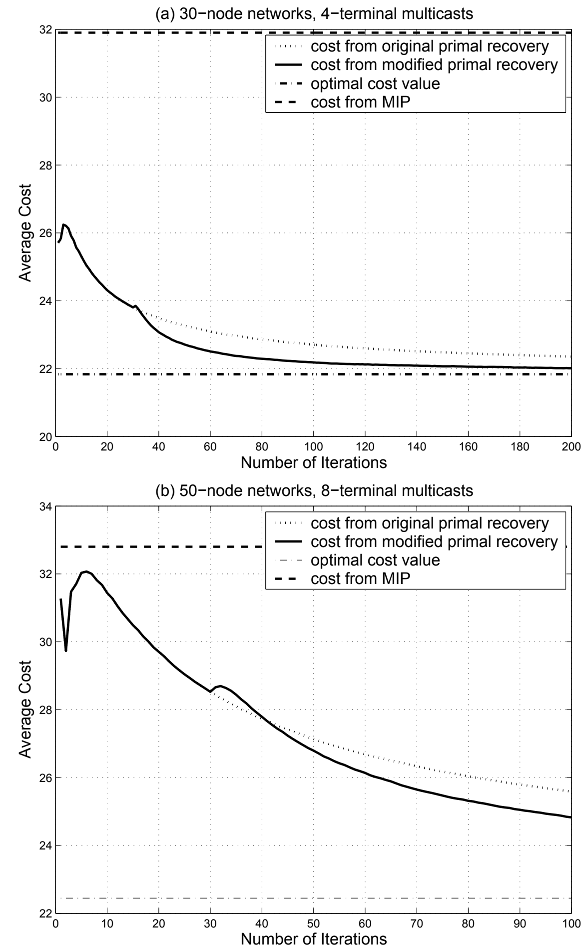

Fig. 2. Average cost of random 4/8-terminal multicasts in 30/50-node wireless networks, using the decentralized subgraph optimization algorithms and centralized MIP algorithm. For modified primal recovery method, $N_a = 30$.

Direction Mobility Model [15], where each node selects a random direction between $[0,\ 2\pi]$ and a random speed between $[\mathsf{minspeed}, \mathsf{maxspeed}]$. A node travels to the border of the simulation area in that direction, then randomly chooses another valid direction and speed, and continues the process. Note that our algorithms are applicable to all mobility models, and we have chosen this specific one for its simplicity.

In our studies, we assume that the nodes are traveling at a speed that is slow relative to the node computation speed and link transmission rate. Under such assumptions, we consider the movement of the nodes in small discrete steps, and between each step, the set of links in the network and their costs are considered constant. We refer to the period between two discrete steps as a "static period", and let the number of subgraph optimization iterations performed within each static period be N_s.

We ran simulations for the various methods presented in Section III. For dual variable initialization, we only present results based on the projection method, since it gives the

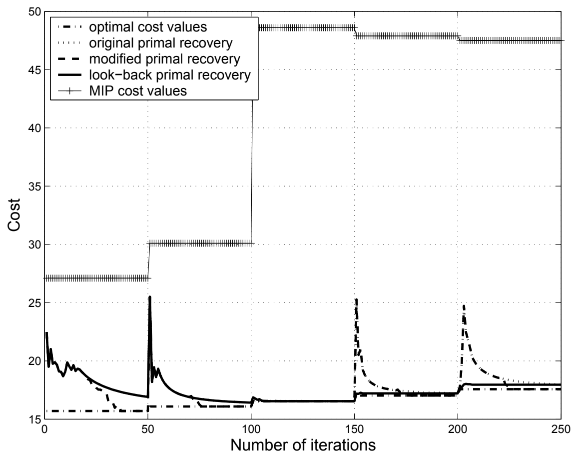

Fig. 3. Cost of a random 4-terminal multicast in a 30-node mobile wireless network, with $N_s = 50$, under various algorithms. For the modified primal recovery method, we used $N_a = 20$ and, for the look-back primal recovery method, we used $N_a = 50$.

best performance. Here, each node has a random speed in the interval $[0, 0.1]$ units/static period. We choose this range because the steps taken by the nodes with such speeds are relatively small as compared to r, and our assumption that the network is static between steps is valid. Also, this is a relative speed of the nodes with respect to the static period, and we can vary N_s to simulate different actual speeds of the nodes.

To illustrate the typical performance of the subgraph optimization scheme in a mobile wireless network, Fig. 3 shows the costs for each iteration for an instance of the multicast problem. As expected, if we flush the memory of $\{\hat{x}[n]\}$ at the end of each static period, and start accumulation for the primal cost afresh, the cost of the multicast is very spiky. On the other hand, if old $\{\hat{x}[n]\}$ values are used when they are feasible, the primal cost is usually much smoother. Of course, if node movement renders the old $\{\hat{x}[n]\}$ values infeasible, we have no choice but to start afresh, and the curves for original primal recovery and look-back primal recovery coincide (as in the 2nd static period in Fig. 3).

In Fig. 4, we show simulation results under different network and multicast settings, and for nodes with different speeds. The parameter N_a used in the modified primal recovery is set to 20. First, we compare the performance of the three options to recover primal solutions. Under the same settings, look-back primal recovery gives the lowest average cost, followed by modified and original primal recovery. We also observe that when the same methods are used, the faster the node moves, the higher the average primal cost is, owing to the lack of time for the algorithm to converge. Also, a network with more nodes or a multicast with more terminals makes convergence of the decentralized algorithm slower, and thus results in higher average primal cost.

The simulation results have shown that the decentralized subgraph optimization scheme is robust in mobile wireless networks when the nodes are moving slowly relative to the

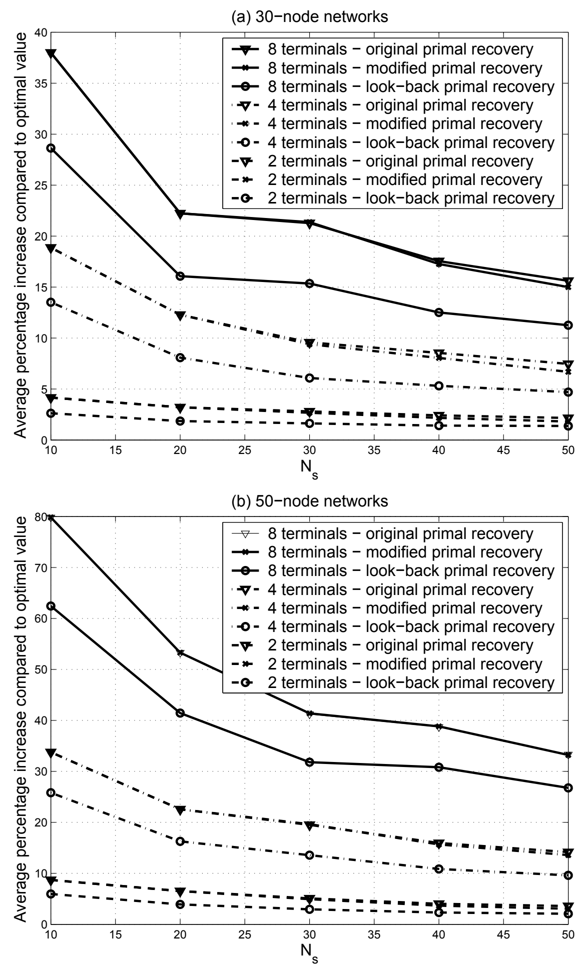

Fig. 4. Extra energy required for multicasts in mobile wireless networks using decentralized subgraph optimization scheme in terms of percentage of the optimal value.

computation and message exchange rate of the nodes. On average, it can track the changes in the optimal value closely, and in most cases, requires lower energy for multicast than MIP even though the nodes are mobile and computation is done at each node in a distributed manner.

V. CONCLUSION

Subgraph optimization is an important problem in coded networks. In this paper, we studied the subgradient method for performing this optimization in a decentralized manner. Simulation results show that the subgradient method produces significant reductions in multicast energy as compared to centralized routing algorithms after just a few iterations. Moreover, we have shown that the algorithm is robust to changes in the network and can converge to new optimal solutions quickly as long as the rate of change in the network is slow as compared to the speed of computation and transmission. These favorable results demonstrate that the subgradient method is a promising foundation for a protocol for transmission over coded networks.

Acknowledgment

This work was supported by the National Science Foundation under grants "ITR: Network Coding - From Theory to Practice" (CCR-0325496), "XORs in the Air: Practical Wireless Network Coding" (CNS-0627021), and by the Air Force Office of Scientific Research (AFOSR) under grant FA9550-06-1-0155.

References

[1] D. S. Lun, M. Médard, T. Ho, and R. Koetter, "Network coding with a cost criterion," in *Proc. 2004 International Symposium on Information Theory and its Applications (ISITA 2004)*, Oct. 2004, pp. 1232–1237.

[2] T. Ho, M. Médard, R. Koetter, D. R. Karger, M. Effros, J. Shi, and B. Leong, "Toward a random operation of networks," submitted to *IEEE Trans. Inform. Theory*.

[3] T. Ho, M. Médard, J. Shi, M. Effros, and D. R. Karger, "On randomized network coding," in *Proc. 41th Annual Allterton Conference on Communication, Control, and Computing*, Oct. 2003.

[4] T. Ho, R. Koetter, M. Médard, D. R. Karger, and M. Effros, "The benefits of coding over routing in a randomized setting," in *Proc. 2003 IEEE International Symposium on Information Theory (ISIT 2003)*, 2003.

[5] P. A. Chou, Y. Wu, and K. Jain, "Practical network coding," in *Proc. 41th Annual Allterton Conference on Communication, Control, and Computing*, Oct. 2003.

[6] T. Ho, B. Leong, M. Médard, R. Koetter, Y.-H. Chang, and M. Effros, "On the utility of network coding in dynamic environments," in *Proc. 2004 International Workshop on Wireless Ad-hoc Networks (IWWAN 2004)*, 2004.

[7] Y. Wu, P. A. Chou, and S.-Y. Kung, "Minimum-energy multicast in mobile ad hoc networks using network coding," submitted to *IEEE Trans. Commun.*.

[8] D. S. Lun, N. Ratnakar, R. Koetter, M. Médard, E. Ahmed, and H. Lee, "Achieving minimum-cost multicast: A decentralized approach based on network coding," in *Proc. IEEE Infocom 2005*, vol. 3, Miami, FL, Mar. 2005, pp. 1608–1617.

[9] R. Ahlswede, N. Cai, S.-Y. R. Li, and R. W. Yeung, "Network information flow," *IEEE Trans. Inform. Theory*, vol. 46, no. 4, pp. 1204–1216, July 2000.

[10] R. Gowaikar, A. F. Dana, R. Palanki, B. Hassibi, and M. Effros, "On the capacity of wireless erasure networks," in *Proc. 2004 IEEE International Symposium on Information Theory (ISIT 2004)*, Chicago, IL, June–July 2004.

[11] D. S. Lun, M. Médard, R. Koetter, and M. Effros, "Further results on coding for reliable communication over packet networks," in *Proc. 2005 IEEE International Symposium on Information Theory (ISIT 2005)*, Sept. 2005, pp. 1848–1852.

[12] D. S. Lun, M. Médard, and R. Koetter, "Efficient operation of wireless packet networks using network coding," in *Proc. International Workshop on Convergent Technologies (IWCT) 2005*, June 2005, invited paper.

[13] H. D. Sherali and G. Choi, "Recovery of primal solutions when using subgradient optimization methods to solve Langrangian duals of linear programs," *Oper. Res. Lett.*, vol. 19, pp. 105–113, 1996.

[14] J. E. Wieselthier, G. D. Nguyen, and A. Ephremides, "Energy-efficient broadcast and multicast trees in wireless networks," *Mobile Networks and Applications*, vol. 7, pp. 481–492, 2002.

[15] T. Camp, J. Boleng, and V. Davies, "A survey of mobility models for ad hoc network research," *Wirel. Commun. Mob. Comput.*, vol. 2, no. 5, pp. 483–502, Aug. 2002.